

Real-Time Cartesian Force Feedback Control of a Teleoperated Robot *

Perry Campbell
Lockheed Engineering and Sciences Company
Houston, Texas 77058

Abstract

Active cartesian force control of a teleoperated robot is investigated. An economical micro-computer based control method is tested. Limitations are discussed and methods of performance improvement suggested. To demonstrate the performance of this technique, a preliminary test was performed with success. A general purpose bilateral force reflecting hand controller is currently being constructed based on this control method.

I. Introduction

Most available hand controllers today do not have force reflectance capability, which gives the human operator much needed force information. Such a capability is needed in order for operators to better control their manipulators. The systems which do use force feedback are made of dedicated sub-systems which cannot be inter-changed, i.e. the hand controller from one system is not useable with the robot of another system. For example, the force-reflecting hand controller manufactured by Kraft, Inc. is not compatible with a PUMA or Robotics Research robot. Work at the Teleoperator Systems Branch Laboratory of the Engineering Directorate at the Johnson Space Center is currently addressing this problem as it relates to laboratory and space teleoperators.

The objective of this work is to construct and demonstrate a force reflecting hand controller which is capable of driving several different types of robot slaves. The hand controller is currently under construction, so only preliminary system performance data is available at this time.

In order for such a hand controller to be compatible with different types of slave robots, it must perform its control function in some coordinate frame which is common to most manipulators. The most obvious frame to use is a cartesian (tx,ty,tz,rx,ry,rz) frame which is either fixed in the end link or in the base of the robot. Many robot controllers are factory programmed to accept cartesian position commands with no modifications to their control software. Most non-force reflecting hand controllers today are designed to issue command signals in cartesian space.

* This work presents the results of work carried out at the Johnson Space Center, under contract No. NAS9-17900, National Aeronautics and Space Administration.

The "common module" concept has been successfully applied in the Shuttle program. For example, the Shuttle Remote Manipulator System (RMS) has a translational rate hand controller and a rotational rate hand controller. Both of these hand controllers are identical to those used to control the Shuttle's navigation jets, with the exception of an additional switch installed on the rotational hand controller (for rate hold of the RMS). In the Shuttle, the primary advantage is that there was no re-design effort for the RMS hand controllers. Another important factor, which is only now beginning to be appreciated, is that one-of-a-kind items are very susceptible to premature obsolescence. This effect is being observed in the Shuttle program, where original vendors are now beginning to close plants which manufacture parts which are used only by a particular shuttle sub-system. It is very costly to obtain new parts when this happens (a new manufacturer must be contracted to second-source the hardware). When a particular part is used in many places on the Shuttle, a single programmatic effort is all that is required to assure a continuing supply of flight spares throughout the life expectancy of the vehicle. It is expected that commonality will be an important factor in sub-system component design in the Space Station program. There will be several different types of robotic sub-systems on or around the Space Station Freedom, each of which may require a hand controller signal. If these sub-systems utilize force feedback in the teleoperator mode of operation, then some type of force reflecting hand controller will be needed for each teleoperator sub-system. Clearly, an attempt should be made to utilize a single design for all of these applications. For the Space Station, there is the additional factor of the duration of its mission: for extended periods of space activity, the risk is greater that a hardware failure will render a system useless unless spare components are kept on-board (very expensive, but sometimes necessary). If some or all of the teleoperator sub-systems aboard the Space Station used a common hand controller, then a single flight spare could replace any one of them, should it fail.

There is also a need for a single hand controller which can operate several geometrically different robots in our robotics laboratory. We want to have the capability to re-run a test using any of the robots, rather than being required to use only the robot for which a specialized hand controller was built. A single generic hand controller is obviously less expensive than several custom-built hand controllers. There will also be a need for a general purpose hand controller in the MPAC (Multi-Purpose Applications Console), which is a generic operator control station also being developed under the Engineering Directorate.

Several tests have been planned in the Telerobotics Laboratory related to this project. Two are of importance here:

1. Cartesian force control of a single robot.
2. Cartesian force feedback control of a robot and a hand controller.

Of these two tests, only test number 1 has been completed. Test number 2 is pending hardware construction. This report will present an overview of the results of test number 1 and will discuss the present state of test number 2.

II. Hardware Configuration

In test number 1, a PUMA 762 robot was programmed to perform motions in the External Alter control mode. The commands sent to the robot were computed by an 80286 based personal computer (PC-AT). The inputs for the PC-AT computations

were a set of forces and torques from a force / torque sensor (F/TS) mounted at the robot tool plate. Behavior characteristics were programmed on the PC so that the robot tip would respond to forces as desired. Several different behaviors were coded and tested and the performance of the system was studied for each case. (Figure 1.)

In test number 2, a PUMA 762 or Robotics Research (RR) 2107 or 1607 will perform as a slave robot under the control of the above-described hand controller. (Figure 2.) The hand controller is to be constructed from a PUMA 250 or 260 robot. This is possible because bilateral force reflectance requires that both master and slave perform similarly. It is sometimes helpful to visualize such a system as two robots being controlled together, rather than one robot and one hand controller. At one robot, the environment it interfaces with is a human operator (a hand). At the other robot, the environment is the actual environment which the operator wishes to interact with.

The present hardware configuration requirement for test 2 is as follows:

1. Large robot and controller (PUMA 762 or RR 1607).
2. Small robot and controller (PUMA 250 or 260).
3. PC-AT with STARGATE serial interface hardware.
4. Cabling as necessary.
5. Two force / torque sensors.

Test number 1 required only one robot and one force / torque sensor.

III. Results of Test Number 1

In test number 1 the function of the control PC was solely to wait for the VAL controller to request another position command, send a command when requested, request data from the Force Torque Sensor every time the VAL asks for a position command, and to read the F/TS data when it arrives on the serial port.

The demonstration given for test 1 utilized several very simple control laws. One law implemented made the end point behave as if it were the center of gravity of a mass in a zero-g field. (a free body). The physical law governing a free body is: $F=m*a$. Therefore the acceleration of the body should be proportional to the external force acting on it. The external alter function of VAL expects the command signal to be either position or change of position (cummulative or non-cummulative). The latter mode was used in this test. The velocity of a free body is simply the integral of the acceleration. Digitally, it is the sum of all previous force measurements divided the sampling period and also divided by the mass we wish to make the end point behave like.

Another control law implemented was similar to the above law except a certain amount of damping was added, making the "free body" behave as if it were moving through a viscous fluid. This was done by using the same law as above and subtracting a term proportional to the previous velocity command.

A third law was to make the end point behave like a massless damper. This was the simplest law to implement because the velocity command is simply proportional to the F/TS measurement. The end point would move at a rate which was exactly proportional to the force at the end point.

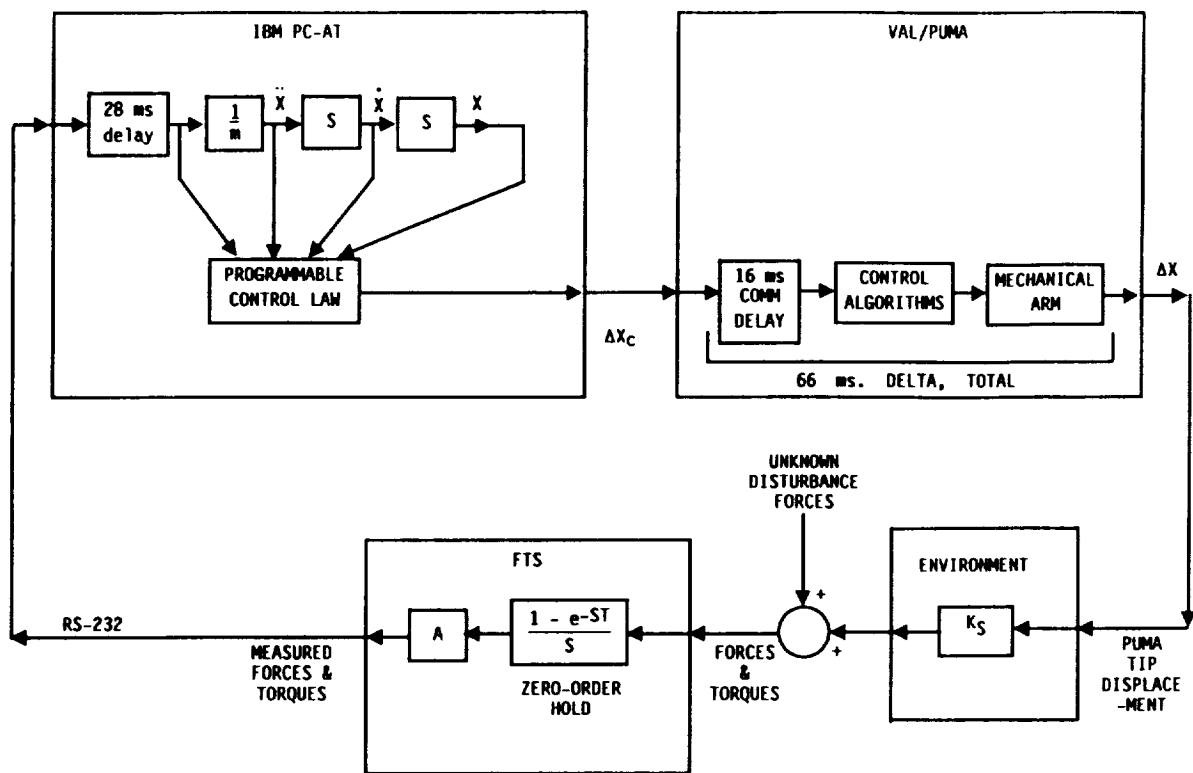


Figure 1: Force Feedback Control Diagram.

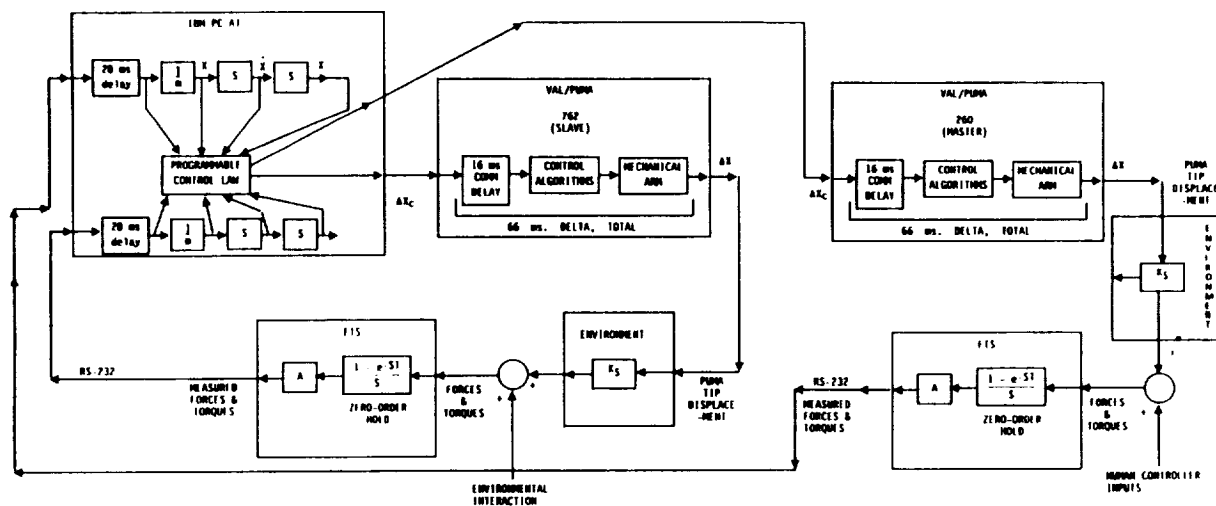


Figure 2: Force Reflecting Hand Controller Block Diagram.

During the evaluation of the above control law algorithms, it was discovered that momentum was not being conserved in the "free body". This was seen when a bias force was added and a spring mounted under the gripper. The arm tip was allowed to "fall" down to a table-top. If the end point was behaving like a free mass (with the simulated g-field) it should bounce back up to the same height from where it was dropped. This was not the case. The end point bounced up significantly higher on each bounce. Thus, momentum was not being conserved in our simulated environment. The reason for this behavior is speculated to be:

The Puma arm does not reach the commanded position until about 66 ms. after the command is issued. The net effect is that the contact point of the arm is allowed to continue to travel into the table-top for a full 66 ms. longer than it should be allowed to. This delay allows the spring to be compressed a full 66 ms. longer than it would have been if a true mass had been used. Thus, when the tip exits in the opposite direction, the exit velocity is larger than when it entered because the measured forces are larger than they would have been had a true mass been used. This effect has been confirmed by simulation of the cartesian control system.

A simple effort was made to reduce the above described effects. A predictive model was added to the PC software which read the current forces and used them and the previous forces to linearly extrapolate what the forces were going to be when this position command was actually reached by the PUMA arm. This made the system stable but still did not conserve momentum. This is because the predictions are required to be exact for conservation of momentum to hold and our prediction model is not exact. Nevertheless, the predictive model made the end point behave much more like a free body than without it. Other methods (such as correcting the estimate after the fact) were deemed beyond the scope of this test.

Another problem associated with the delays and sampling time was the inability of the arm to maintain a constant force on a rigid surface. This was because the two hard surfaces tended to bounce at rates much faster than could be compensated for. This caused a problem when the two rigid surfaces were to be held together at a constant force. By adding the predictive algorithm and a large amount of damping, the problem was resolved for the case of laying the gripper directly on a table top. Much stiffer surfaces could still cause problems. The modifications made caused the manipulator to behave very sluggishly. The sampling rate used in this test was about 40 Hz.

It is anticipated that sampling rates of about 500 Hz will be sufficient for most space-based force feedback manipulator control systems.

IV. Test Number 2 - Accomplishments to Date

In test 2, the control algorithm in the PC takes 2 force/torque measurements and issues 2 cartesian position commands to the robot controllers during each update period. The system delays are about twice that in test 1 and so performance is expected to be diminished by about a factor of two. This test has not been performed yet. To date, the only data available are expected results, taken from a computer simulation of the proposed hardware/software arrangement.

Preliminary simulation indicates that a reasonable performance is attainable. Figure 3 illustrates the force-tracking capability of the simulated controller in response to a sinusoidal input at one robot end effector while the other robot is constrained in a spring-like environment. Figure 4 shows the step response of

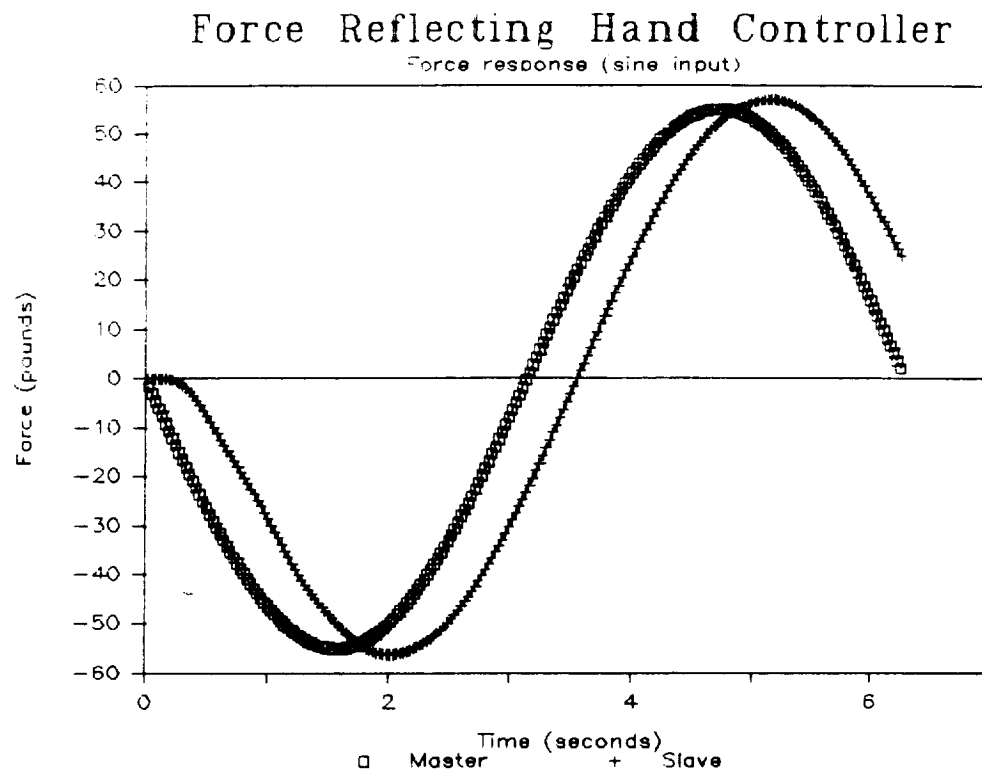


Figure 3: Simulated Force Tracking (Sinusoidal Input).

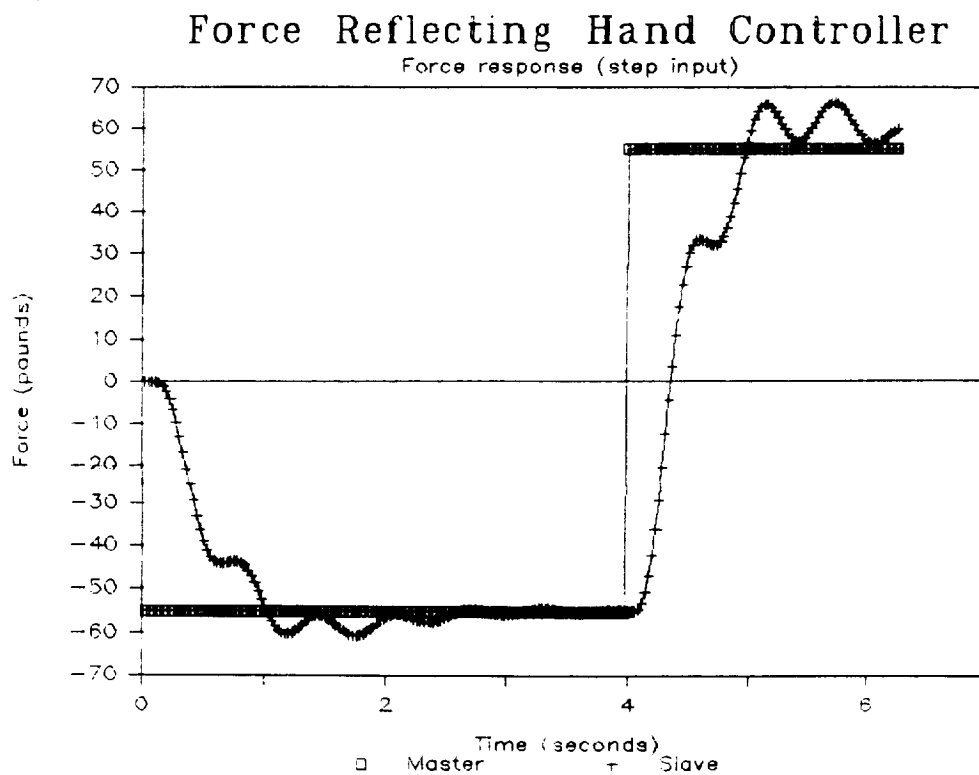


Figure 4: Simulated Force Tracking (Step Input).

the same system. It is assumed that the manipulator performs as a perfect positioning device. Each robot and controller is modelled as a perfect positioner with a mass attached to it by a spring. The mass interacts with the environment, which is also modelled as a spring. The control algorithm simulated is as follows, for each time step:

$$\begin{aligned}
 MF_m(t) &= MK*(MX_{cmd}(t)-MX_{ee}(t)) \\
 MFe(t) &= C*u(t) && \text{(Step response input)} \\
 MA_{ee}(t) &= (MF_m(t)+MFe(t))/M_m \\
 MV_{ee}(t) &= MV_{ee}(t-T)+MA_{ee}(t)*T \\
 MX_{ee}(t) &= MX_{ee}(t-T)+MV_{ee}(t)*T \\
 MX_{cmd}(t) &= MX_{cmd}(t-T)+(MF_m(t)-SF_m(t))*T*MG \\
 SF_m(t) &= SK*(SX_{cmd}(t)-SX_{ee}(t)) \\
 SFe(t) &= -MX_{ee}(t)*MKe && \text{(Spring-like environment)} \\
 SA_{ee}(t) &= (SF_m(t)+SFe(t))/S_m \\
 SV_{ee}(t) &= SV_{ee}(t-T)+SA_{ee}(t)*T \\
 SX_{ee}(t) &= SX_{ee}(t-T)+SV_{ee}(t)*T \\
 SX_{cmd}(t) &= SX_{cmd}(t-T)+(SF_m(t)-MF_m(t))*T*SG
 \end{aligned}$$

where:

A preceding 'M' denotes a Master quantity
A preceding 'S' denotes a Slave quantity
e.g. MF_m = Master force (measured)
 SF_m = Slave force (measured)

F_m = force (measured)
 F_e = force (at environment interface)
 A_{ee} = acceleration of robot tip
 V_{ee} = velocity of robot tip
 X_{ee} = position of robot tip
 X_{cmd} = commanded position of robot tip
 T = time step update period (28 ms. for PUMA)
 t = present time
 m = end effector mass
 C = arbitrary magnitude for step response input
 K_e = environmental stiffness at contact point
 K = effective stiffness of robot at tool
 G = cartesian controller gain

This controller essentially sets the velocity of each of the robots to be proportional to the difference between the measured forces. Simplicity is the driver for selecting this algorithm. It is clear how this control architecture is well suited to other control algorithms such as active stiffness control [1], impedance control [2], and hybrid control [3].

V. Conclusions

From these results, it appears feasible and even desirable to construct a bilateral force reflecting hand controller based on cartesian force control algorithms. System performance is mainly limited by computer power. The system under development is limited by the lower level robot controller loop times. This limitation may be extended greatly by modifying the individual robot controllers. Update rates of 40 Hz make the system too sensitive to the environment. It is estimated that 200 to 500 Hz will be more acceptable. A plan has been established to upgrade the Robotics Research controller to obtain a looptime of approximately

2 ms. Such computational demands can be met by a high end dual-processor 80386 based personal computer with math coprocessors or by a reduced instruction set computer such as the IRIS 4D/70G.

REFERENCES

1. Salisbury, J.K. (1980) "Active Stiffness Control of a Manipulator in Cartesian Coordinates". Proceedings of the 19th IEEE Conference on Decision and Control, pp. 95-100.
2. Hogan, N. (1985) "Impedance Control: An approach to Manipulation: Part I - Theory, Part II - Implementation, Part III - Applications". Journal of Dynamic Systems, Measurement, and Control, pp. 1-24.
3. Craig, J.J. & Raibert, M.H. (1981) "Hybrid Position/Force Control of Manipulators". Transactions of ASME, Journal of Dynamic Systems, Measurement, and Control, Volume 102, pp. 126-133.